Machine Learning and Adaptation of Domain Models to Support Real Time Planning in Autonomous Systems

A. Introduction and Research Context

Background: Simulating *low-level* cognitive behaviour, such as reaction to stimuli or autonomic activity, has been a major focus of research and development in the autonomous systems (AS) community for many years. Automated assessment of sensor data, and reactive action selection in the form of condition-action pairs, is well developed in robotic and control application areas. In contrast, a characteristic of *high-level* cognitive behaviour is the ability to reason with knowledge of action and change in order to synthesise plans to achieve desired long term goals. This area is not so well understood, or manifested in applications of real time dynamic AS. Utilising such reasoning abilities enables an agent to choose which action to perform to achieve a desired task based on a deliberative process involving knowledge of the environment, resources, goals, and available actions. The implementation of such high-level behaviour has been considered problematic in the AS community in the past, regarding both the real time reasoning and knowledge representation aspects as intractable [34].

Control systems in autonomous vehicles, however, such as in exploration robots or space satellites, have to be capable of deliberative planning and scheduling (P&S) to autonomously accomplish high-level tasks (e.g. collect a rock sample at position X, take a photograph of constellation Y). In fact, scientists at NASA for over 20 years have been developing systems with such P&S technology for the control of autonomous vehicles, and have deployed systems which can plan the control of spacecraft, generate activities for uploading to spacecraft, schedule observation movements for the Hubble Telescope, and control underwater vehicles [4, 20, 6, 18]. Research into this kind of deliberative planning is often termed artificial intelligence (AI) P&S. The AI P&S research community has been successful in overcoming some of the theoretical problems to do with computational complexity of generative planning, and scale-up of proposed solutions, which dogged the community in the last century. This is evidenced by the deployment of AI P&S technology in a wide range of applications: at this year's annual ICAPS event[1] the fielded applications reported included fire fighting, satellite control, emergency landing, aircraft repair scheduling, workflow generation, narrative generation, and battery load balancing. The event also hosts competitions leading to the development of optimised planning tools which can be embedded in applications software.

[1] icaps11.icaps-conference.org

Challenges of Fielding AI Planning: The basic challenges of utilising symbolic reasoning systems such as deliberative planners within real time AS are well known, and were neatly summarised some time ago by Woolridge and Jennings [34]:
(a) the transduction problem: that of translating the real world into an accurate, adequate symbolic description, in time for that description to be useful; (b) the representation/reasoning problem: that of how to symbolically represent information about complex real-world entities and processes, and how to get agents to reason with this information in time for the results to be useful.
This work and similar publications in the agent community discouraged approaches using symbolic reasoning, although hybrid approaches have been explored in for example dynamic environments [31], and multi-agents systems [13].

The reasoning problem alluded to in (b) is what many in the AI P&S community are aiming to solve, and a measure of their success is the growing range of applications alluded to above. It is expected that this ongoing research will lead to yet more efficient solvers, which can accept more expressive input languages. More fundamental, and the subject of this proposal, is the "transduction" problem in (a), which is connected to the representation issue of (b).

For an artificial agent to produce plans and decisions rationally, it has to have knowledge of the objects and the dynamic effects of actions within its environment. A symbolic representation of such knowledge is called a *domain model*, and seperation of the concerns of creating a domain model, and the creation of a planning algorithm, is the basis of what is termed *domain independent planning*. This is in contrast to specialised or fixed goal planning such as path planning, where the separation of knowledge of the domain and planning algorithm is often blurred. Acquiring, validating and maintaining a domain model for the purposes of automated reasoning is a key research challenge, and has long been a limiting factor in the exploitation of domain indepenendemt planning. Currently domain models are hand crafted and maintained, whereas in AS they are required to be automatically learned and subject to adaptation over run time. The aim of this project is to draw on recent research advances in AI P&S in working towards overcoming this research challenge, expressed in the **research hypothesis**: *Automatically learning and maintaining an accurate and adequate domain model for the purposes of high-level reasoning, in particular for the processes of P&S, enables effective, sustained goal-directed behaviour for real time dynamic AS.*

By the end of this project we aim to have demonstrated with a prototype the feasibility of real time deliberative planning in AS, based on a self-adapting domain model. If this challenge is achieved, then it will open the door to implementing high-level cognitive behaviour in real time dynamic AS.

We next survey the state of the art, focusing on *adequacy*, that the expressiveness of current domain model languages, and *accuracy*, in particular the use of automated techniques to form and keep up to date the domain model in the context of verification and validation constraints.

Domain Model Languages: The control mechanisms of ASs need to be able to represent and reason with rich and detailed knowledge of such phenomena as movement and resource consumption in the context of uncertain and continuously changing environmental conditions [12]. Traditionally, physical systems with discrete and continuously-varying aspects have been represented using the mathematical notion of a *hybrid* dynamical system. This is a system that has a state made up of a set of real and discrete- valued variables that change over time according to some fixed set of constraints. Hybrid systems are used for modelling in applications such as embedded control systems [5].

The research-led standard domain model language in planning is PDDL (planning domain description language), which is based around a world view of parameterised actions and states, where it is assumed that a controller generates a collection of instantiated actions to solve some goal posed as state conditions. It has been extended to cope with real applications such as crisis management [9] and workflow generation [26], and has versions which can represent time and resources. More expreesive modelling languages such as PDDL+ have been developed for applications where reasoning about processes and events in a mixed discrete/continuous world is necessary [10]. PDDL+ was recently used in an application for developing multiple battery usage policies [19]. Although PDDL is designed for logical precondition achievement, specialist forms of planning can be incorporated into the language using *procedural attachment* [8]. Using this kind of mechanism low level planning procedures such as real time path planning, which benefits from a range of specialist techniques[21], can incorporated within PDDL.

Despite its widespread acceptability, a serious problem with PDDL is that it reflects the concerns of those working in **generative** planning, rather than the execution and scheduling orientation of many applications. In contrast, scientists at NASA Ames developed the application-oriented language families HSTS [22] and then NDDL [17] for their applications in the Space arena. NDDL is fundamentally different to PDDL in that encodings are based around representations of objects and object instances, which persist in predefined timelines of continuous activities. Each activity has a start and end time interval (to represent uncertainty of duration), and the distinction between *action* and *state* is effectively blurred. Plan generation and execution are therefore linked to a much greater degree than with PDDL. NDDL's concept of timelines are related to the idea of crafting abstract plans as in the input languages to HTN systems [16]. The idea of pre-written hierarchical plans to formulate possible behaviours has long been a popular type of formalism in which to encode dynamic knowledge for AI applications. A related view of how one could formulate dynamics comes from the area of Cognitive Robotics [25], which also seeks to emphasise the integration of planning and execution. The idea here, though, is to start with an axiomatisation of the application environment using a variant of situation logic, then hand craft generic plans (so-called 'action programmimg') from which concrete plans can be efficiently derived using deduction. Systems used in Cognitive Robotics such as GOLOG require more engineering for individual applications than in classical planning, but appear more appropriate for the control of robotics devices.

Another strand of research, closely linked to HTN and practical planning, has focussed on *rich plan representations* [23, 29, 30]. These representations are intended for the sharing of plans between agents. The richness of these languages stems from the underlying ontology that contains generic concepts from the planning domain. They have been used in a number of application domains such as emergency response [24] and personnel recovery [33].

The common role of these rich and expressive language families is to enable engineers to formulate an adequate representation of structural, dynamic and heuristic knowledge for applications involving action and change. In real time autonomous systems these languages have been used to represent a high level knowledge layer. *The key limitation here is the hand coded nature of this kind of knowledge, and the difficulty of validating the model* - all current applications rely on teams of knowledge engineers to encode and validate the domain model [15]. To meet the challenge of domain modelling in NDDL, recent work by NASA scientists is aimed at developing an interactive domain model editor (IMDE) which uses a simulator to short circuit the loop between the model and validation of the model [3]. This work also points to the use of machine learning techniques (some developed by the authors of this proposal) to assist in engineering the model. Another promising method that can be used to automatically synthesise a planning domain model is to translate from an existing formal model in an *application lanaguage*. The ICKEPS-09 competition was devoted to this area, with applications including e-Learning, web services composition, and business processing [27]. While this line of work is important in the context of embedding planning components in applications such as workflow planning, this is not so suitable for AS where no formal model exists a priori. Also, in AS the domain model is subject to refinement and adaptation over time, in order that goal directed planning function will remain effective. We propose to adopt machine learning techniques to effect both the *initial acquisition* of the domain model, and its evolution over its lifespan.

Machine Learning of Domain Models: Machine Learning applied to AI P&S has attracted a long history of research, and we point the reader to a recent survey for a full account [14]. There have been many events on the subject in recent years including workshops adjunct to AI international conferences (including ICAPS), and elements of the ICAPS competition series (ICKEPS/IPC). In the context of domain indepenedent plannng, as well as research aimed at learning a domain model representing the physics of the world, much of the machine learning work is aimed at learning heuristics to make the use of a planning engine more efficient.

Domain model learning can be separated into three concerns: (i) what language is the learned domain model going to be expressed in? (ii) what inputs (training examples, observations, constraints, partial models etc) are there to the learning process? (iii) what stage is the learning taking place - initial acquisition, or incremental, online adaptation? For most work done up to now the answers to (i) are "some variant of PDDL forming a domain model that can be input to planning engines" and to (iii) is initial acquisition. However, adaptation can be viewed as a special case of initial acqisition, where input to the learning process includes the current domain model as well as training examples etc, and output is the updated model.

Regarding (ii), systems that learn very expressive domain models tend to demand most detailed input. Work in learning domain models for robotic agents [1, 2] assumes that a training mechanism exists with rich feedback mechanisms. Typically, much a priori knowledge is assumed, such as predicate descriptions of states, and partial or total state information before and after action execution. With such rich inputs, systems such as Amir's SLAF [1] can learn actions within an expressive action schema language.

Some recent work on learning domain model has concentrated on learning with little or no supplied domain knowledge. The LAMP system [36] can form simple PDDL domain theories from example plan scripts and associated initial and goal states only. It inputs object types, predicate specifications, and action headings, and from plan scripts taken from planning solutions, it learns a domain model. The domain model is synthesised using a constraint solver, inputting two sets of constraints: one set is based on assumed physical, consistency and teleologial constraints - for example, every action in the example plan script adds at least one precondition for a future action, actions must have non-empty effects, and so on. The other set of constraints is generated using a type of associative classification algorithm which uses each plan script as an itemset, and extracts frequent itemsets to make up constraints. While LAMP is aimed squarely at helping knowledge engineers create a new new domain model, LOCM is an algorithm learns from plan scripts only [7]. As with ARMS, it outputs a planning domain theory in a PDDL format but it inputs *only* plan scripts - it does not require representations of initial and goal states, or any descriptions of predicates, object classes, states etc. LOCM has been used in a system that learns to play the Freecell game by observation, with no a priori knowledge of the game [7].

There have been several other notable developments in learning in uncertain or partially known domains. *Reinforcement learning*, traditionally used in single goal or policy learning planners, has recently been developed for symbolic or relational learning, though its potential for learning full models of the PDDL variety is not yet proven[14]. A promising approach towards learning incomplete and uncertain domain models is ongoing in the *Model-lite* project [35]. Here the authors use probabilistic logic as the basis for the language of the learned domain model.

B. Summary of Aims and Objectives

To summarise, before AS in real time dynamic applications can attain high-level cognitive skills there are still major challenges to be overcome in the acquisition, validation and adaptation of domain knowledge. To be able to perform deliberate reasoning in new or changing domains, we propose that an AS needs to be able to learn and incrementally adjust its understanding of the world, encapsulated in a domain model. It needs to ensure the accuracy of the evolving domain model with the help of internal verification checks and external validation constraints. The project aims to work towards the solution of these challenges within a programme involving collaborator applications (identified as *CAs* below) put forward by collaborators in the consortium behind the AIS program call. Hence, we set up the following objectives, the achievement of which will be measured using the criteria following each one:

1. *develop an expressive, hybrid domain model language (here called AIS-DDL) for AS*
   Criteria: AIS-DDL will be a generic language applicable to the CAs, containing required feautes such as mixed discrete/continuous domain knowledge. It will be capable of capturing knowledge about actions and change at a human-understandable level of abstraction, and allow for efficient reasoning as required for learning, planning and validation.

2. *research and develop methods for automated domain model acquisition and online adaptation*
   Criteria: the methods will be generic to the CAs, with output consisting of models in AIS-DDL; they will maintain the accuracy and adequacy of the domain model, and develop heuristic knowledge to support planning functions;

3. *determine methods and develop tools for knowledge analysis, verification and validation*
   Criteria: the methods will be able to detect inconsistencies in the learned models, derive new knowledge, and inform further knowledge acquisition and learning cycles. Further, V&V criteria will be in terms humans can understand, thus enabling a mixed initiative approach to knowledge engineering where appropriate.

4. *deliver a prototype demonstrator system*
   Criteria: The system will exhibit deliberative planning within the CAs in a virtual world, and therein demonstrate the efficacy of domain model acquisition and online adaptation.

Fit with the AIS programme call: This proposal will advance the state of knowledge in four areas of the Research Interests table:

- Model Building and Learning (8.0). The proposal concerns the acquisition, learning, validation, maintenance and adaptation of "reference models" (here called domain models)

- Planning (4.0). The main role of the domain model (referred to in the previous point) is to enable automated planning to achieve desired goals.

- Structural Awareness and Information Abstraction (3.0). To be able to adapt and change the domain model requires information inferred from sensor data.

- Verification and Validation of Autonomous Systems (7.0). The proposed project will contribute to this in so far as the V&V of the domain model and learned knowledge.

This proposed project's research is seen by the proposers as fundamental to *all* the collaborators' scenarios as described in the Call.

C. Method and Technical Plan

This research project's method will be based around the following activities:
– the creation/acquisition of a simulation environment tailored to each CA (analogous to that proposed by Scientists from NASA/JPL [3] to explore mixed initiative knowledge engineering). This will provide the necessary environment for investigative research for domain model learning and verification and validation of learned knowledge and synthesised plans.
–utilising a hierarchical approach to domain model construction, with abstract symbolic knowledge at a high level to enable long term planning, with detailed knowledge at a lower level to enable path planning or manipulator planning
–the creation of verification axioms and processes based on the ontological constraints intrinsic to the design of AIS-DDL (analogous to those developed for PDDL [32])
–the engineering of a set of immutable validation constraints capturing the physics of the application domain
–utilising existing domain model learning tools such as LOCM and LAMP referred to above, and the rich sources of relevant literature, for example the international workshop series on P&S for space[2].

With these developments in place, it will be feasible to meet the main challenge of domain model learning. The learned high level knowledge in AIS-DDL will be translated into the input language of existing planning engines in order to test generated plans using simulation, and the simulator will be used as the basis of the subsequent demonstrator system.
Project Risks: We identify major risk areas in the project as (a) feasibility of creating simulations of CAs (b) poor degree of fit between planing technology and the application requirements (c) difficulty in obtaining and eliciting underlying knowledge. The range of potential CAs (as demonstrated in the program call) and the similarity of them to existing planning developments (eg Mars Rover) mitigate against (a). The vast experience of the Proposers in applying AI P&S, and in the knowledge engineering aspects in general, will help resolve problems arising in (b) and (c) by judging what is feasible in terms of the scope and range of the CAs given the timescale. Finally, the project plan is arranged flexibly into six work packages which are progressive and self contained, meaning that deliverables, which will have an external impact, are output at each stage of the project.

[2]http://www.congrexprojects.com/11c05/

WP1. Analysis of CAs and State of the Art: Determination and analysis of requirements of the set of CAs which cover the high level planning and decision making function of the AS, drawn from members of the AIS consortium. Scope of CAs, and identification of experts, documents and other resources available to be used. For each CA: detemination of required planning function, collation of sample required plans, state representations and sensor information.
Distill the state of the art from the literature as applicable to the case studies. Acquisition and testing of applicable tools eg specialist and general planners, learning tools, with potential for use in the project.
Construction of project web site and consideration of routes to transfer technology and exploit research outputs. Consideration of potential for integration of project results with other funded research in the AIS programme.
Delivered: Agreements on the detail and scope of the CAs, such as I/O from/to a deliberative planning function, and a set of detailed criteria with which to measure success [D1]; a collection of literature and summary overview of applicable state of the art in planning and learning techniques[D2], a repository of potentially applicable research tools, project website, and initial report on the integration of research results within the AIS programme[D3].
Evaluation: Scope of CAs to be sufficiently testing to measure all the planned features of the domain model language, the learning method, online adaptation, validation etc. The survey will be of publishable standard, and the tools repository will be used to demonstrate to collaborators the potential of current real time planning and learning technology.

WP2. Configuration of Simulation Environment: Using D1, D3 and collaborator resources where applicable, configure or acquire a simulation environment, for example based on a virtual world platform (such as Second Life), to simulate CAs. Identification of the abstractions made and the effort required to transfer systems developed in the virtual world to a real scenario.
Delivered: report on abstractions made in the virtual world[D4]; working application simulator, and well defined interfaces [D5],
Evaluation: simulator configured to showcase the chosen CAs, the execution of plans based on learned domain models, and handling user interaction during

execution; visualisation to satisfy end users

WP3. Planning Domain Model Representation and Ontology: Utilising D2, gain insights from the major AI approaches to domain model representation (e.g. in classical planning, action programming, constraint-based planning), and formalisms used in hybrid systems design [5], SAT-based mixed discrete/continuous systems [28], classical-based formalisms [10], and situation-calculus-based work [11]. Clarify the relationship between high level notations and low level reactive planning knowledge as used in the CAs, and specify a generic I/O language for the planning component. Combine with insights from D1 and ceate the first version of AIS-DDL.
– Define a rich ontology of domain independent planning concepts for representing processes, events, actions, uncertainty, and continuously changing variables that will provide the abstract vocabulary for AIS-DDL;
– Design and implement algorithms that maps AIS-DDL to known langauges such as variants of PDDL to utilise state of the art planning technology.
Delivered: specification of generic planner I/O [D6], AIS-DDL[D7], specification of domain model language ontology[D8], translators[D9].
Evaluation: D6 and D7 will fit the requirements of the planning function and model represention (respectively) of the CAs (evaluated by hand encodings of collaborator problem domains). D8 will be evaluated by peer reviewed publication and in combination with D9 using dynamic testing (in WP4 and WP5).

WP4. Verification and Validation: This WP will research and develop methods and tools for the verification and validation of AIS-DDL domain models, resulting in more accurate and robust domain models, and a way of validating the doman model learning processes(WP5). The work will draw on D6,D7 and D8 and relevant literature [32, 15, 16], and produce tools for
a) automated verification analysis: the creation of verification axioms and processes based on the ontological constraints intrinsic to the design of AIS-DDL
b) automated validation checks: the engineering and encoding of a set of immutable validation constraints capturing the physics of each of the CAs
c) a visualisation tool to allow users to validate by inspection and manipulate the domain models
The outputs of these tools will be used as follows:
– to provide additional input knowledge during the knowledge acquisition process, and to inform each cycle of domain model adaptation;
– to output information relevant for the efficiency with which planners can solve planing problems, and to provide advice on the best planner to use, and to help optimize the representation to support efficient automated planning.
– to augment learned models with knowledge useful for the human user (to make them more understandable and intelligible), and useful for enabling translation to other formalisms;
– derive new knowledge in terms of domain independent features that provide further insights into the underlying model.
Delivered: verification tool [D10], validation tools[D11], visualisation tool [D12], report on specification and computational properties of tools[D13]
Evaluation: D10-D12 will be evaluated taking into account number of errors identified from test scenarios, the quality of the additional knowledge created, and the success in integrating the output with learning functions in WP5, D13 will be submitted for peer review.

WP5. Machine Learning and Adaptation of Domain Models Utilise D2 and D3 to further investigate forms of knowledge acquisition and learning, and methods for domain model creation. Assemble a number of sources of input to machine learning, for each of CAs: (i) sets of sample information fused from sensor data (ii) domain invariant information (iii) derived information from D10,D11 in WP4. Utilise KE tools from D3 as appropriate to create sample domain model encodings for the CAs. Utilising D7 (the planning ontology), and insights from the literature e.g. [36, 7]
a) create an initial domain model acquisition tool
b) based on a), create an adaptation tool for evolving the domain model through its online use
Delivered: Hand crafted domain models[D14], learning[D15] and adaptation[D16] tools, report on specification and computational propoerties of the tools[D17]
Evaluation: Learned domain models will be compared to D14; the process of adaptation of domain models will be evaluated operationally within the demonstrator(WP6), D17 will be sent for peer review publication.

WP6. Demonstrator Systems, Project Evaluation and Exploitation: Development of the simulation environment to incorporate autonomous behaviour in order to demonstrate system learning and adaptation capabilities; extensive testing using CAs scenarios; overall evaluation of project; potential for future development including integration with other results in the AIS porgamme, identification using D3 of effort need to transfer results from the virtual world to the real, and determination of exploitation routes of developed technologies.
Delivered: final versions of simulation environments and demonstrator scenarios[D18]; pathway to research exploitation document[D19]; final project report[D20]
Evaluation: evaluate D18, D19 against success measures identified in D1 and take up of research results by commercial partners; peer reviewed journal publications derived from D20.

References

[1] E. Amir. Learning partially observable deterministic action models. In *Proc. IJCAI 05*, pages 1433–1439, 2005.

[2] S. Benson. *Learning Action Models for Reactive Autonomous Agents.* PhD thesis, Stanford University, Palo Alto, California, 1996.

[3] Bradley J. Clement, Jeremy D. Frank, John M. Chachere, Tristan B. Smith, Keith Swanson. The challenge of grounding planning in simulation in an interactive model development environment. In *Proc. KEPS Workshop, ICAPS*, pages 23–30, 2011.

[4] J. L. Bresina, A. K. Jónsson, P. H. Morris, and K. Rajan. Activity planning for the Mars Exploration Rovers. In *Proc. ICAPS*, pages 40 – 49, Monterey, California, USA, 2005.

[5] L.P. Carloni, R. Passerore, A. Pinto, and A. Sangiovanni-Vincentelli. *Languages and tools for hybrid systems design*. 2006.

[6] Steve Chien, Benjamin Smith, Gregg Rabideau, Nicola Muscettola, and Kanna Rajan. Automated planning and scheduling for goal-based autonomous spacecraft. *IEEE Intelligent Systems*, 13:50–55, September 1998.

[7] S.N. Cresswell, T.L. McCluskey, and Margaret M. West. Acquiring planning domain models using LOCM. *Knowledge Engineering Review (To Appear)*, 2011.

[8] P. Eyerich, T. Keller, and B. Nebel. Combining Action and Motion Planning via Semantic Attachments. In *Proc. ICAPS*, 2010.

[9] J. Fdez-Olivares, L. Castillo, O. Garcia-Perez, and F. P. Reins. Bringing users and planning technology together: experiences in SIADEX. In *Proc. ICAPS*, pages 11 – 20, Cumbria, UK, 2006.

[10] M. Fox and D. Long. Modelling mixed discrete-continuous domains for planning. *Journal of Artificial Intelligence Research*, 27:235 – 297, 2006.

[11] H. Grosskreutz and G. Lakemeyer. On-line execution of cc-golog plans. In *Proc. IJCAI*, pages 12 – 18, 2001.

[12] J. Bresina, R. Dearden, N. Meuleau, S. Ramakrishnan, D. Smith and R. Washington. Planning under Continuous Time and Resource Uncertainty: A Challenge for AI. In *Proc. Conference on Uncertainty in Artificial Intelligence*, 2002.

[13] Rune Jensen and Manuela M. Veloso. Interleaving deliberative and reactive planning in dynamic multi-agent domains. In *In Proceedings of the AAAI Fall Symposium on on Integrated*, pages 22–24. AAAI Press, 1998.

[14] Sergio Jiménez, Tomás De la Rosa, Susana Fernández, Fernando Fernández, and Daniel Borrajo. A review of machine learning for automated planning. *The Knowledge Engineering Review*.

[15] D. Long, M. Fox, and R. Howey. Planning domains and plans: Validation and analysis. In *Proceedings of the Verification and Validation in Planning workshop, ICAPS'09*, 2009.

[16] T. L. McCluskey, D. Liu, and R. M. Simpson. GIPO II: HTN Planning in a Tool-supported Knowledge Engineering Environment. In *Proc. ICAPS*, 2003.

[17] C. McGann. How to solve it: Problem solving in Europa 2.0. Technical report, NASA Ames, 2006.

[18] C. McGann, F. Py, K. Rajan, J. Ryan, and R. Henthorn. Adaptive control for autonomous underwater vehicles. In *Proc. AAAI*, pages 1319–1324. AAAI Press, 2008.

[19] M.Fox, D.Long, and D.Magazzeni. Automatic Construction of Efficient Multiple Battery Usage Policies. In *Proc. ICAPS, Frieburg, Germany*, 2011.

[20] G. E. Miller. Planning and scheduling the hubble space telescope: Practical application of advanced techniques. In *Artificial Intelligence, Robotics, and Automation for Space Symposium*, pages 339 – 343, 1994.

[21] M.Naveed, A.Crampton, D.Kitchin, and T.L.McCluskey. Real-Time Path Planning using a Simulation-based Markovian Decision Process. In *31st SGAI International Conference on AI (to appear)*, 2011.

[22] N. Muscettola. HSTS: Integrating planning and scheduling. In *Intelligent Scheduling*, pages 169–212. Morgan Kaufmann, 1994.

[23] A. Pease and T. Carrico. Object model working group core plan representation. Technical Report AL/HR-TP-1996-0031, United States Air Force Armstrong Laboratory, Wright-Patterson AFB, OH, 1996.

[24] S. Potter, A. Tate, and G. Wickler. Using I-X process panels as intelligent to-do lists for agent coordination in emergency response. In *Proc. 3rd Information Systems for Crisis Response and Management (ISCRAM)*, 2006.

[25] R. Reiter. *Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001.

[26] A. Riabov and Z. Liu. Scalable planning for distributed stream processing systems. In *Proc. ICAPS*, Cumbria, UK, 2006.

[27] Roman Bartak, Simone Fratini, and Lee McCluskey. The third competition on knowledge engineering for planning and scheduling. AI Magazine, Spring 2010, 2010.

[28] J.A. Shin and E. Davis. Processes and continuous change in a sat-based planner. *Artificial Intelligence*, 166, 2005.

[29] A. Tate. Roots of SPAR—shared planning and activity representation. *The Knowledge Engineering Review*, 13, 1998.

[30] A. Tate. <I-N-C-A>: A shared model for mixed-initiative synthesis tasks. In Gheorghe Tecuci, editor, *Proc. IJCAI Workshop on Mixed-Initiative Intelligent Systems*, pages 125–130, 2003.

[31] A. Walczak, L. Braubach, A. Pokahr, and W. Lamersdorf. Augmenting bdi agents with deliberative planning techniques. In *in The Fifth International Workshop on Programming Multiagent Systems (PROMAS-2006*, 2006.

[32] G. Wickler. Using planning domain features to facilitate knowledge engineering. In *Proc. KEPS Workshop, ICAPS*, 2011.

[33] G. Wickler, A. Tate, and J. Hansberger. Supporting collaborative operations within a coalition personnel recovery center. In *Proc. 4th Knowledge Systems for Coalition Operations (KSCO)*, pages 14–19, 2007.

[34] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[35] S. Yoon and S.Kambhampati. Towards model-lite planning: A proposal for learning & planning with incomplete domain models. In *Proc. Workshop on AI Planning and Learning, ICAPS*, 2007.

[36] Hankz Hankui Zhuo, Qiang Yang, Derek Hao Hu, and Lei Li. Learning complex action models with quantifiers and logical implications. *Artificial Intelligence*, 174(18):1540 – 1569, 2010.